

The development of a GUI program for automated generating rotor blade airfoil performance table[†]

Taewoo Kim¹, Kwanjung Yee^{1,*}, Sejong Oh¹ and Hee Jung Kang²

¹*Department of Aerospace Engineering, Pusan National University, Busan, 609-735, Korea*

²*Rotor Team, Korea Aerospace Research Institute, Daejeon, Korea*

(Manuscript Received March 5, 2008; Revised July 2, 2008; Accepted August 18, 2008)

Abstract

A user-friendly, Windows-based Graphic User Interface (GUI) program that automates the generation of the aerodynamic performance tables for comprehensive helicopter load analysis codes was developed. With this program, computational grids are automatically created for the given airfoil surface coordinates, and users can verify the grids immediately. In addition, the program's post-processing feature can enable the real-time inspection of the interim results of the computation. The aerodynamic performance tables can be generated by performing automated aerodynamic analysis of the airfoil in various angles of attack and Mach numbers by using validated CFD code. The Mixed-Language technique was employed to combine the FORTRAN-based CFD codes to the C++-based GUI program without needing any transformation between these two languages. The MS Access was used to construct the data base of the aerodynamic tables. The real-time inputs of the analysis results of the ODBC Access data base make it easy to retrieve the aerodynamic tables. Of the entire range of the angle of attack (-180° to 180°), the CFD analysis is only performed in the significant angles depending on the Mach number. The aerodynamic data for the other angles were obtained by using an interpolation method; therefore, the tables could be generated quickly. Various aerodynamic information such as drag divergence Mach number and stall angle can be extracted from the computed results stored in the database.

Keywords: Computational fluid dynamics; Airfoil aerodynamic analysis, Graphic user interface; Automatic analysis; Rotor blades; Airfoil data interpolation; Data-mining

1. Introduction

In the prediction of the aerodynamic load for a helicopter rotor blade, it is critical to consider not only the aerodynamic forces themselves but also the structural deformation by these forces, because rotorcraft blades usually deform more easily than the wings of a fixed-wing aircraft. Though the ability to separately predict the vibration load and the aeroelastic characteristics has been significantly improved due to the achievements associated with the researches in the CFD and structural dynamics for the last decades, it still takes a

long time to complete an aerodynamic load analysis. This is because a complex flow field exists as a result from the effect of the unsteady rotor wake [1]. Since the analysis time is becoming a scarce commodity, the comprehensive analysis codes turn out to be extremely useful for helicopter load analysis in the practical design process. Most of the current comprehensive rotor analysis codes require 2-D airfoil aerodynamic performance tables because these codes often utilize the blade element theory which needs 2-D airfoil aerodynamic data instead of 3-D flow field information [2]. Additional corrections on the 2-D data in the comprehensive codes should be made to reflect the effects due to sweep angle, unsteady flow, compressibility and finite wing, etc. Most industries that design helicopters acquire those data through systematic wind tunnel

[†] This paper was recommended for publication in revised form by Associate Editor Yang Na

* Corresponding author. Tel. +82 51 510 2481, Fax: +82 51 513 3760

E-mail address: daedalus@pusan.ac.kr

© KSME & Springer 2009

during the computation, 4) store the real-time data through ODBC, or 5) generate aerodynamic data for the entire range of angle of attack by using the resulting data of the significant angles of attack. The grid generation involves an operation used to interpolate the missing airfoil data. In addition, it is possible to verify the airfoil shape and the result of CFD computation by using the program. The flowchart of the program is depicted in Fig. 2.

2.2 The automated generation of the airfoil grid

2.2.1 The airfoil shape interpolation using the cubic spline method

If the number of user-specified nodes is not sufficient, the coordinate should be interpolated to define the proper shape before the grid for CFD analysis is generated. If the connectivity is an important factor needing to be considered, the cubic spline interpolation is a more desired alternative used to connect remote points than the other methods such as linear interpolation, Lagrange interpolation, and Newton polynomial interpolation methods [4].

The cubic spline interpolation method uses a cubic polynomial equation between two adjacent data points. This equation has four independent variables and thus requires four boundary conditions to be solved. Two of the solutions come from the assumption that the line should include two adjacent points. The other two can be derived from the condition that the first and the second derivatives of the points should both be continuous [5].

The cubic polynomial equation for the *i*th interval between two arbitrary points is given by Equation (1):

$$y = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \quad (1)$$

Eqs. (2) to (5) can be obtained by solving Eq. (1) and applying the proper boundary conditions.

$$b_i = \frac{S_i}{2} \quad (2)$$

$$a_i = \frac{S_{i+1} - S_i}{6h_i} \quad (3)$$

$$d_i = y_i \quad (4)$$

$$c_i = \frac{y_{i+1} - y_i}{h_i} - \frac{2h_i S_i + h_i S_{i+1}}{6} \quad (5)$$

where, S_i, S_{i+1} are the second derivatives at points (x_i, y_i) and (x_{i+1}, y_{i+1}) respectively.

To verify the cubic spline interpolation method, three airfoils, NACA 0012, RAE 2822 and VR 13,

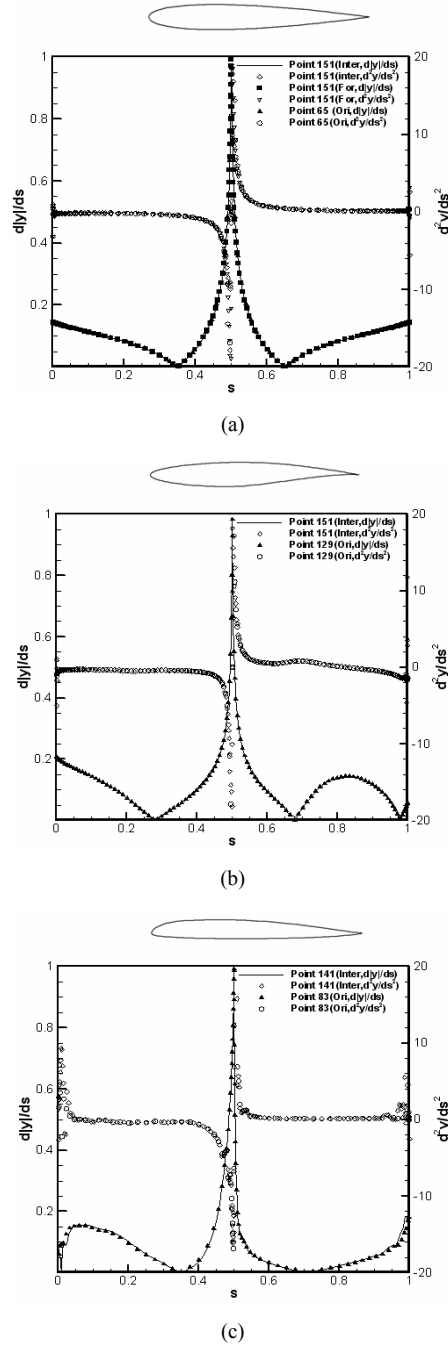


Fig. 3. Comparison of the airfoil shape interpolation (a) NACA0012, (b) RAE2822, (c) VR13.

with different shape and curvature were used to evaluate the accuracy of the coordinates before and after the interpolation in terms of the variation of the y value per unit length of the surface. Since the data of the NACA0012 airfoil was directly obtained from the formulation, the interpolated results were compared with the data of the NACA0012 airfoil to verify the cubic spline interpolation method. Fig. 3 shows the comparison results of the first and the second accuracy for three different airfoils. The first accuracy is the first derivative calculated by the absolute value of y of the airfoil surface data and the distance between the data points, while the second accuracy is the second derivative by the y value of the surface data. ‘Inter’, ‘Ori’, and ‘For’ represent the approximate data calculated by Eq. (1), the input surface data, and the data calculated by NACA airfoil formula, respectively.

The result shows that the accuracies of the airfoil data are almost identical for most regions. For the airfoils with the non-closed trailing edges, its trailing edges are extended to the vertices to generate the C-type grids. In such case, when the values of the new points are greater than 1, the points from 70% chord moved toward the leading edge. For VR13 airfoil, the result shows that interpolating the open-airfoil data to the closed-airfoil data moved the data toward the leading edge.

2.2.2 Hyperbolic grid generation

The interpolated airfoil shape is fed into the hyperbolic grid generation program to create a C-type CFD analysis grid. Assuming the computational domain coordinates are defined as (r, s, t) and the total length of the computational domain is defined as 1, it leads to $\Delta r = \Delta s = \Delta t = 1$. If the coordinate of the surface is given as $x_0(r, s) = x(r, s, t = 0)$, $x(r, s, t)$, it can be found by hyperbolic partial differential equation. The tangential resultant and normal governing equations for the grid are given by Eq. (6) and (7), respectively [6].

$$x_r(r, s, t) = \frac{x_r(r, s, t) \times x_s(r, s, t)}{\|x_r(r, s, t) \times x_s(r, s, t)\|^2} \Delta V \quad (6)$$

$$= \frac{\Delta V}{\|x_r(r, s, t) \times x_s(r, s, t)\|} n(r, s, t)$$

$$n(r, s, t) = \frac{x_r(r, s, t) \times x_s(r, s, t)}{\|x_r(r, s, t) \times x_s(r, s, t)\|} \quad (7)$$

The grid spacing along the normal direction de-

pends on clustering methods. For the grids with the total extent D and the number of the nodes N, the clustering can be done by using Eqs. (8) and (9).

$$d_i^n = \frac{D}{N} \text{ Constant-distance clustering} \quad (8)$$

$$d_i^n = D \alpha^n \frac{\alpha - 1}{\alpha^{N+1} - 1} \text{ Constant-ratio clustering} \quad (9)$$

The volume of the cell can be given as the average cell area multiplied by the spacing as follows:

$$\Delta V_i = d_i^n \frac{\Delta \alpha_i}{\Delta \alpha_i} \quad (10)$$

The automated generation of the grid is done by using the GUI program depicted in Fig. 4. The program displays the user-specified airfoil data on the screen, takes inputs from the dialog window for the hyperbolic grid generation, and finally generates the grid automatically. The generated grid is displayed on the screen so that the user can verify it immediately.

2.3 Analysis scheme of the CFD code

A 2-D Navier-Stokes code, which was applied and validated in the study of Ref. [7], is used for the aerodynamic analysis. The governing equation for the code is given by Eq. (11)

$$\frac{1}{J} \frac{\partial Q}{\partial t} + \frac{\partial E}{\partial \xi} + \frac{\partial F}{\partial \eta} = \frac{\partial E_v}{\partial \xi} + \frac{\partial F_v}{\partial \eta} \quad (11)$$

where, $Q = [\rho \ \rho u \ \rho v \ \rho e \ k \ w]^T$, $J = \xi_x \eta_y - \eta_x \xi_y$, t_c is the pseudo computational time, t is the physical time. The length, density, and speed are non-dimensionalized by the chord length C , the air density ρ_∞ , the speed of sound α_∞ , respectively.

Roe’s Flux Difference Scheme is used for the inviscid terms, and the central difference scheme is used for the viscous terms. For the time integration, the dual time stepping and the data parallel symmetric Gauss-Seidel (DP-SGS) are used. The dual time stepping technique iterates at the pseudo time Δt_c with respect to the fixed physical time and converges the solution considering the physical time term in the right hand side of Eq. (11) as the source, and DP-SGS is suitable for parallel computation. In addition, Menter’s shear stress transport (SST) model is employed as the turbulence model.

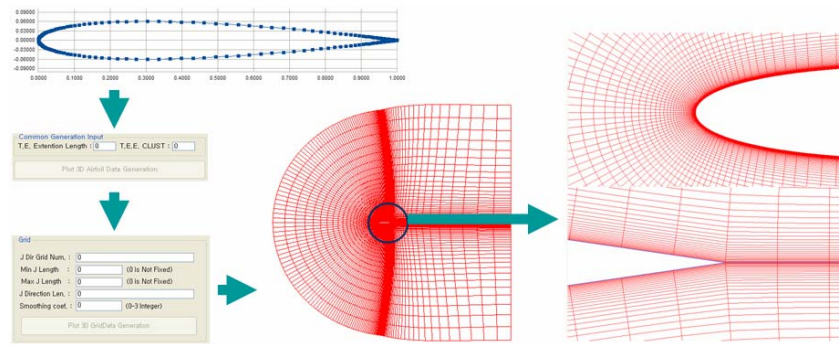


Fig. 4. GUI for the automated grid generation.

SST is known to be a relatively accurate model for the 2-D flow with massive separation such as dynamic stall.

2.4 GUI program development

Using the powerful graphic features of the Windows operating system, GUI offers programming methods for a user-friendly interface. For the GUI program development, Microsoft provides Microsoft Foundation Class (MFC) programming techniques in Visual Studio. The MFC uses general C++ language with pre-built fundamental classes that construct Windows program components such as application, frame windows, documents, and views, etc. The automated airfoil aerodynamic analysis program in this research was developed by using the MFC [8].

Unlike the Graphic Device Interface (GDI) that the old GUI programs used to compose graphic displays, the GDI+ from the recent Microsoft Developer Network (MSDN) can provide more powerful graphic features in Window XP [9]. The GDI+ is used in the program for detailed display of the grid, including zooming in and out, and panning features.

Though most commercial programs are developed to have GUI, most engineering analysis programs are written in FORTRAN. However, it is very time consuming and intricate to transform the analysis codes written in FORTRAN to GUI-suitable languages such as C++ or Java in order to satisfy the user's requirements.

Currently, reading the output files from the code to the GUI program is widely used as a rudimentary way of connecting FORTRAN codes with GUI programs. In this research, the MSDN-suggested Mixed-Language technique that makes the FORTRAN CFD code a dynamic link library (DLL) is utilized. As a

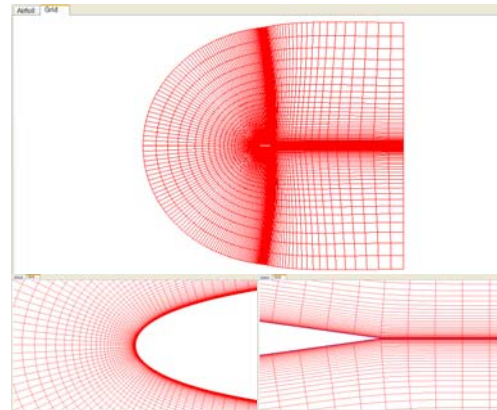


Fig. 5. The displayed grid (Top View) and graphs with zooming in and panning features (Bottom View).

result, it enabled the GUI program to call the FORTRAN code and the subroutines of the code like calling functions, and it also allowed the data exchange to occur in real time.

One issue is that the screen display cannot be updated for the GUI programs that contain loop controls. This is due to the internal control delay for the display update messages while the loop is running. If users need to get the computation information, such as convergence and/or C_p , displayed on the screen while the CFD is in loop calculation, they have to set the background task that collects data in real time. The DLL-transformed CFD code was run as a thread for this purpose.

The ChartFX, a graphic tool developed from the Software FX, was employed to display the convergence check, input airfoil shape, and the C_p on the airfoil surface during the CFD computation process [10]. This commercial tool enabled the real-time graphic processing and led to increased accuracy.

The pressure distribution around the airfoil was displayed through the isogram technique by using the 2-D OpenGL [11]. The OpenGL is a well-known graphic interface which is easily extendible and highly compatible with C++.

MS Access was employed for generating the database for the airfoil aerodynamic tables. Access offers the advantage of using ODBC that Windows has without additional complex linking process. When the code runs, the analysis condition table inside Access stores the folder that contains the computational grid and conditions. At the same time, a new analysis index is created automatically. The analysis index makes the post-processing and distinction from the similar conditions easier. The analysis results are stored in the tables with analysis index after the computation.

2.5 Aerodynamic data generation

Considering the accuracy of the current CFD analysis, it is inefficient to perform CFD analysis to generate aerodynamic data for all the angles of attack. Hence, CFD analysis is only performed for the significant angles of attack shown in Table 1. Based on the results, the characteristic curve is used to get the aerodynamic data for the entire angle of attack range, i.e., from -180° to 180° .

The characteristic equations for estimating the aerodynamic data are given by Eq. (12)–(14) [1]:

$$C_l = A \sin 2(\alpha - \alpha_0) \quad (12)$$

$$C_m = B \sin(\alpha - \alpha_0) - C \sin 2(\alpha - \alpha_0) \quad (13)$$

$$C_d = D + E \cos 2(\alpha - \alpha_0) \quad (14)$$

where the unknown coefficients A, B, C, D, and E can be determined by setting the boundary conditions using the CFD analysis results.

Table 1. The angles of attack for CFD analysis [12].

Mach Number	Angle of Attack
0.1 ~ 0.4	-10° to 22°
0.5	-8° to 20°
0.6	-8° to 16°
0.7	-6° to 12°
0.8	-6° to 10°
0.9	-4° to 8°
1.0	-4° to 6°

2.6 Post-processing

The computed aerodynamic data is stored in the database. It is required to develop additional post-process module which reprocesses the computed data and shows the results of data-mining in summary. The stored aerodynamic analysis results such as C_L , C_D , and C_M are displayed in graphs and specific aerodynamic characteristics such as C_{Lmax} , C_{Dmin} are obtained through data-mining. The obtained aerodynamic data are readily shown in excel-format sheets and graphs without requiring additional post-processing software.

The information provided in the post-processing includes C_L , C_D , C_M vs. AoA (Angle of Attack) and Mach number, C_L vs. C_D , $C_{L1.5}$ vs. C_D , $C_{L\alpha}$ vs. M_∞ , α_{L0} vs. M_∞ , C_{Dmin} vs. M_∞ , C_M vs. M_∞ ($C_L = 0.1$ const), C_{Lmax} vs. M_∞ , X_{ac} vs. M_∞ , X_{cp} vs. AoA. The following explains how to reprocess the computed aerodynamic data.

2.6.1 $C_{L\alpha}$ vs. M_∞

The slope of lift curve $C_{L\alpha}$ is obtained from the lift data of a given airfoil in the database. Since the lift curve is assumed linear before stall, two points are selected to obtain the lift slope in the linear range of lift. The points are chosen to be close to zero angle of attack so that the slope is not influenced by the stall effects.

2.6.2 α_{L0} vs. M_∞

α_{L0} is the angle of attack at which lift coefficient, C_L becomes zero. To this end, two points are selected to obtain the linear function connecting them. Then, the zero-lift angle is determined by finding the intersecting location with the X-axis.

2.6.3 C_{Dmin} vs. M_∞

The C_{Dmin} is sought among C_D values in the database for each given Mach number by using SQL grammar. This method has been validated to have high accuracy and is known to reduce post-processing time.

2.6.4 C_M vs. M_∞ ($C_L = 0.1$ const)

The figure shows the general trend of pitching moment with increasing Mach number and also enables one to find the location of Mach tuck, where the pitching moment increases abruptly due to shock waves on the upper surface. The pitching moments at each Mach number are obtained at the angle of attack where C_L is 0.1.

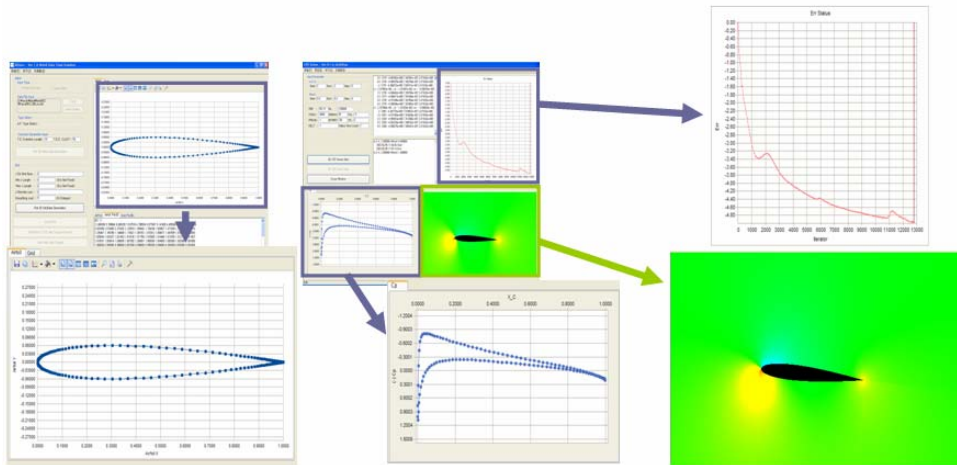


Fig. 6. The post-processing using the chartFX and OpenGL.

2.6.5 C_{Lmax} vs. M_∞

The C_{Lmax} is defined as the maximum lift coefficient before stall for given Mach number. The stall criterion is set such that C_L starts to decrease after the stall angle, which is useful to reduce overall computation time but may cause some inaccuracy when the real maximum exists in the uncomputed region.

2.6.6 X_{ac} vs. M_∞

Variation of aerodynamic centers due to Mach number is calculated from the computed data base. The locations are obtained from the following relations. [1]

$$\overline{X}_{ac} = \overline{X}_a - \frac{dC_{M_\alpha}}{dC_n} \quad (\overline{X}_a = 0.25) \tag{15}$$

$$\frac{dC_{M_\alpha}}{dC_n} = \frac{dC_{M_\alpha}}{d\alpha} \frac{d\alpha}{dC_n} \tag{16}$$

$$C_n = C_l \cos(\alpha) + C_d \sin(\alpha) \tag{17}$$

Here, dC_{M_α} and dC_n are obtained in the stall-free region.

2.6.7 X_{cp} vs. AoA

The centers of pressure due to Mach number variation are obtained from the stored C_L , C_D , and C_M . The locations are determined based on the following relation [1]:

$$\overline{X}_{cp} = 0.25 - \frac{C_{M/4}}{C_n} \tag{18}$$

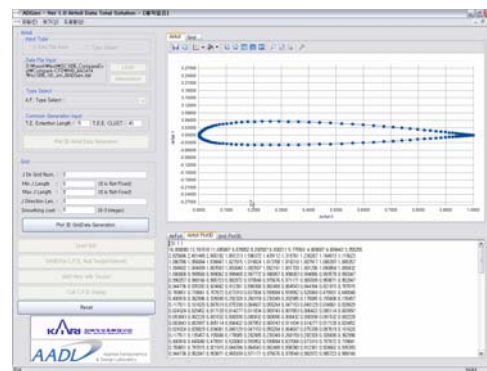


Fig. 7. Airfoil data displayed in the pre-process window.

3. Results

The program consists of two windows: the pre-processing window that takes the airfoil shape input and generates the grid, and the analysis window that performs CFD analysis using the generated grid.

3.1 The Pre-processing window

The pre-processing window takes airfoil surface inputs for analysis and can be used to perform interpolation to ensure a sufficient number of nodes. The airfoil data are displayed on the screen by the ChartFX so that the user can verify the data without additional graphic program. The displayed graph can be zoomed in and out and panned. The text window at the bottom of this window displays the value of the airfoil data for user's convenience, as depicted in Fig. 7.

The trailing edge of the input airfoil was extended,

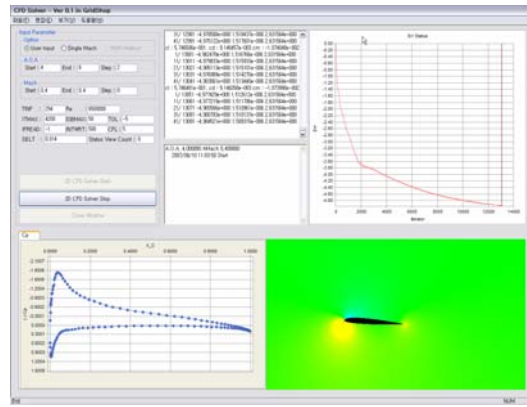


Fig. 8. Analysis condition input/output in the CFD analysis window.

and the C-type grid for CFD analysis was automatically created by using the hyperbolic grid generation method. The input section of the window takes a user's inputs as the initial values. The resulting grid is then displayed on the screen. The zooming and panning features for the graph enable the user to verify the skewness of the grid, and the user can easily verify the grid since zooming the grid does not distort the graph.

3.2 CFD analysis window

The grid generated in the previous step is converted to the grid format that the CFD analysis requires, and the analysis window is activated for the CFD analysis. The analysis window takes the user inputs from the window as shown in Fig. 8. The user can batch various angles of attack and Mach numbers by specifying the range and the intervals so that the program can automatically run the cases without manual operation. The output display interval can be managed as well.

The results can be verified in five different windows. There are two text windows that can show computation results and the information of the angles of attack and Mach numbers when the program is running. The convergence information and C_p are displayed in the graph output windows. As discussed in the previous sections, the use of the ChartFX can improve the graphic output quality. As shown in Fig. 8, the pressure distribution around the airfoil is displayed by the isogram technique using the 2D OpenGL. The post-processing features for data manipulation after the computation will be added in the future.

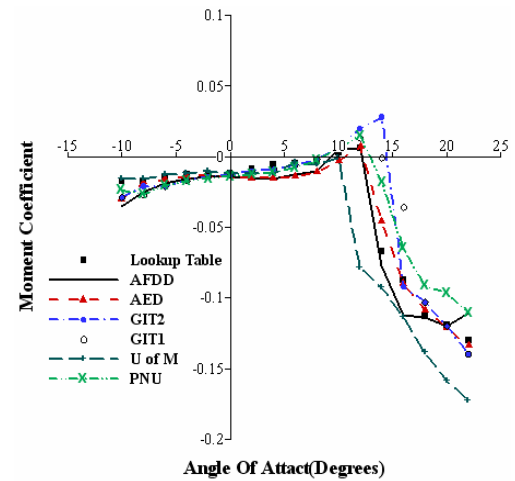
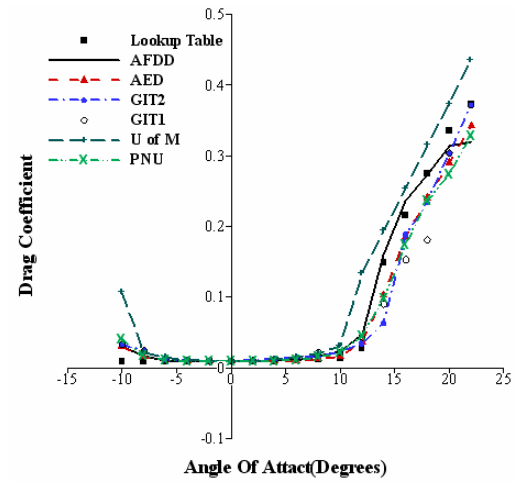
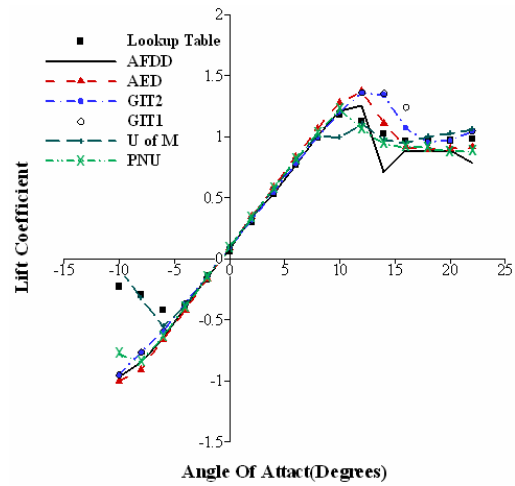


Fig. 9. CFD analysis results comparison for SC1095 airfoil at M = 0.4.

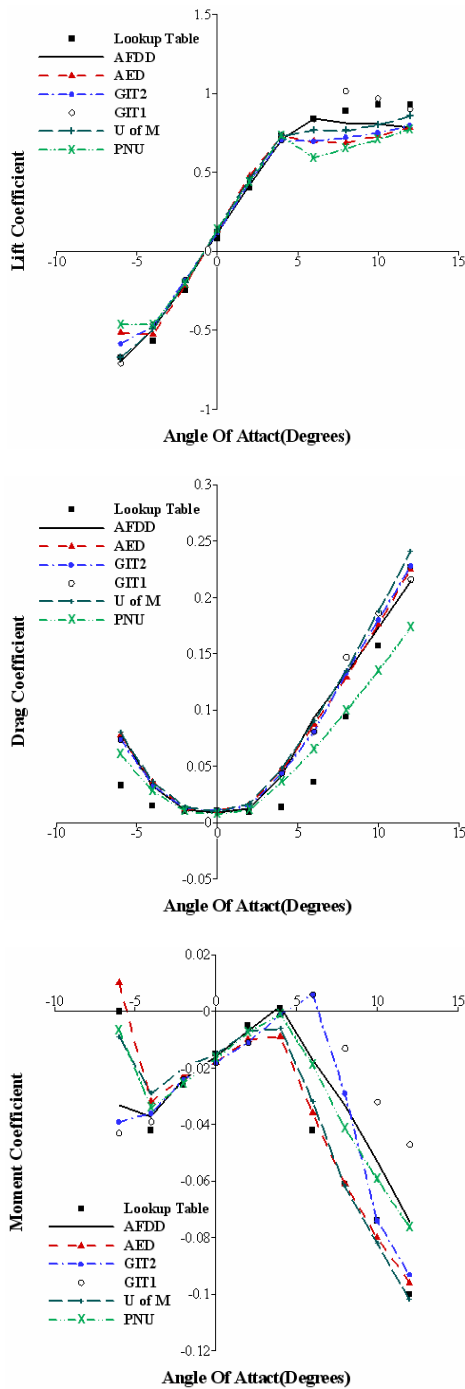


Fig. 10. CFD analysis results comparison for SC1095 airfoil at $M = 0.7$.

3.3 The airfoil aerodynamic table generation

In Figs. 9 and 10, the CFD analysis results obtained from various CFD codes for SC1095 airfoil at Mach

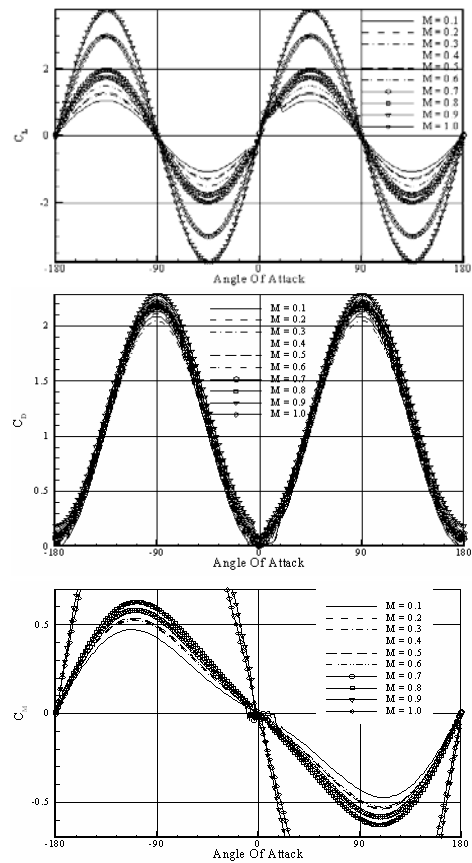


Fig. 11. C_L , C_D , and C_M of SC1095 at the angles of attack from -180° to 180° .

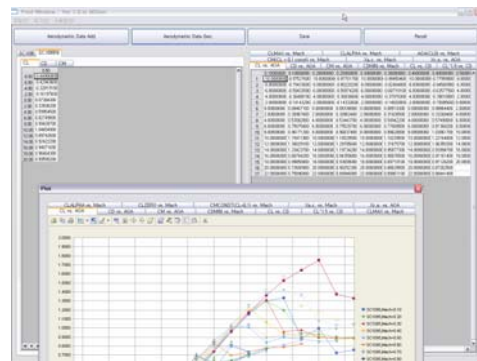


Fig. 12. Post-process window.

number 0.4 and 0.7, $Re\ 6.5e6$ were compared with the CFD analysis results obtained from the research work done by Smith et al. [12]. The CFD codes in comparison include AFDD, AED, GIT1,2, the code from University of Maryland and the lookup table whose values are test results that Sitaraman, Baeder et al. used to

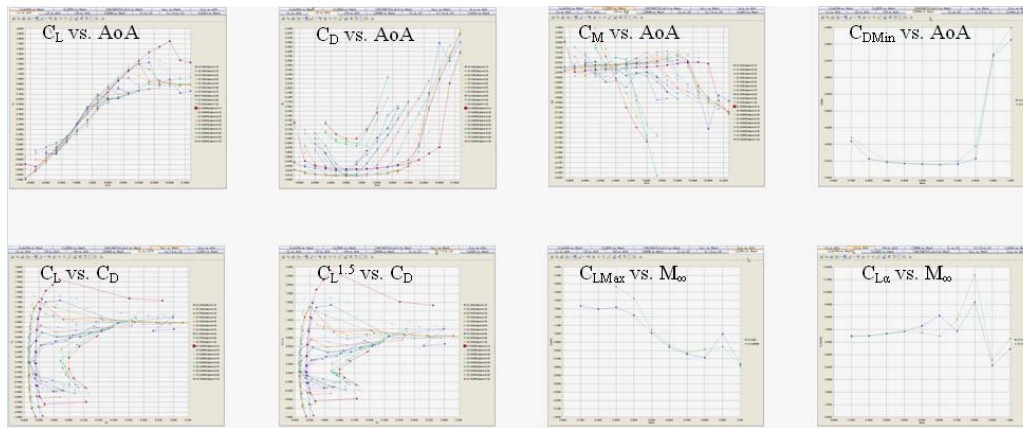


Fig. 13. Data-mining plot.

verify their work [13]. PNU are the results obtained from the automated program developed in this research. Like the other CFD analysis results, the results produced by PNU show good agreement before the stall and some discrepancy after the stall despite the different turbulence models. It can be seen that, even though the user was not involved in the computation process, the PNU analysis is able to produce reliable results.

Fig. 11 shows the plots for C_L , C_D , and C_M from -180° to 180° at Mach number $0.1 \sim 1.0$, $Re\ 6.5e6$. The data used to create these plots was generated by CFD analysis for the significant angles of attack as shown in Table 1. and the characteristic equations for the other angles.

3.4 Post-process window

Besides a grid generation window and CFD analysis window, the program also provides a post-process window which represents the analysis results in summary. All of the required aerodynamic information is shown on the post-process window without additional post-process program. The post-process window displays on screen not only the stored analysis results but also specific aerodynamic characteristics obtained from the post-process mentioned previously.

Fig. 12 shows an example of post-procedure window. As shown, data obtained from data-mining as well as data already stored in the database are displayed in excel format. The results can be readily compared through the windows.

The reprocessed data in the post-processing window can be depicted in spreadsheet as well as in graphical representation. Each aerodynamic coefficient

is separated by tab. Fig. 13 shows an example of graphs provided in post-processing window. The produced graphs are printed by using the tool bar on the window.

4. Discussion and future work

This study presents the development process of an automated program for the aerodynamic analysis of the rotor blade airfoil. Being Linux-based and using script language Perl, the research done by Mayda and Dam [3] had limitations in practical implementation. In addition, the lack of the post-processing feature requires advanced programs to manipulate the data during and after the computation. To resolve the existing issues, a Windows-based GUI program was developed in this research to improve the capabilities of the current program. The program environment offers more user convenience and enhances the practical application because most computers are Windows-based.

The operation of this program is simplified by automated grid generation and the automated execution of the code. The created grid can be displayed on the screen so that the user can verify the grid and perform troubleshooting.

Various cases of CFD analysis can be batch performed by automatically controlling the angle of attack and Mach number instead of manually inputting the data for each case. This feature is suitable for creating aerodynamic tables for the helicopter analysis.

The CFD code that is used in the program is validated in a previous research [7] and the Mixed-Language technique simply connected the code to the

Table 2. Comparison with preceding research.

	Mayda & Dam ('05) [3]	Current Research
OS	Linux	Windows
User Interface	UI, TEXT based	GUI, Graphic based
Language	FORTTRAN, Perl	FORTTRAN, Visual C++
Output Data Type	C81 Format	Supported DB, C81 Format
Post Processing	No	Yes
Result Output	Text Output	Real time Graphic Output

program. The real-time post-processing feature can display the results that the user is interested in.

The steady and unsteady analysis results are obtained to create aerodynamic tables, and then the results are fed to the characteristic equations. The aerodynamic data C_L , C_D , and C_M are stored by using MS Access for easy retrieval and post-processing of the data.

In addition, some useful aerodynamic information could be obtained through data-mining based on the stored computation results.

The present research has several aspects in common with Mayda and Dam [3]'s work. However, considering the everyday OS environment is window-based, this research is thought to have a substantial contribution in terms of usability and ease of use. Table 2. summarizes the major difference between the present work with preceding research. It should be emphasized that the database is utilized first in this program to analyze and summarize the computation results. As shown in Fig. 13, major aerodynamic coefficients are shown in graphical representation using a database, enhancing the ease of the engineer's decision.

Acknowledgment

This study has been supported by the KARI under KHP Dual-Use Component Development Program funded by the MKE, Korea.

References

- [1] J. G. Leishman, *Principle of Helicopter Aerodynamics, 2nd Edition*, Cambridge University Press, (2006).
- [2] W. Johnson, *Helicopter Theory*, Princeton University Press, (1980).
- [3] E. A. Mayda and C. P. van Dam, Automated Generation of Airfoil Performance Tables Using a Two-Dimensional Navier-Stokes Solver, *J. of AHS* 50 (4) (2005) 338-348.
- [4] C. de Boor, *A Practical Guide to Splines*, Springer-Verlag, (1978).
- [5] A. W. Al-Khafaji, J. R. Tooley, *Numerical Methods in Engineering Practice*, Elsevier Science Publishers, (1987) 299-307.
- [6] W. D. Henshaw, *The Overture Hyperbolic Grid Generator User Guide, Version 1.0*, UCRL-MA-134240 (2003).
- [7] K. Yee and W. Joo, Comprehensive Aerodynamic Analysis of a Gurney Flap, *J. of Aircraft* 44 (3) (2007) 1003-1014.
- [8] Y.S. Kim, *Visual C++ 6 perfect guide 2nd Edition*, Youngjin.Com, (2004).
- [9] Microsoft Company, *MSDN Library Visual Studio Oct. 2001*, Microsoft Company, (2001).
- [10] Software FX Company, *ChartFX Client Server API Reference, Ver 5.1 & 6.2*, Software FX Company, Last revised September 23, (2005).
- [11] Silicon Graphics Company, *Redbook: OpenGL Programming Guide, Ver. 1.1*, Addison-Wesley Publishing Company, (1997).
- [12] M. J. Smith et al., Evaluation of CFD to Determine Two-Dimensional Airfoil Characteristics for Rotorcraft Applications. *60th Annual Forum of the American Helicopter Society*, Baltimore, MD, (2004).
- [13] J. Sitaraman, Baeder, J. D., and Chopra, I., Validation of UH-60A Rotor Blade Aerodynamic Characteristics using CFD, *59th Annual Forum of the AHS*, Phoenix, AZ, May, (2003).



Taewoo Kim received a B.S. degree in Aerospace Engineering from Pusan National University in 2007. He is currently a M.S. candidate at the graduate school of Aerospace Engineering at Pusan National University in Pusan, Korea. His research interests are in the area of aerodynamics, optimization design, and helicopter aerodynamics.



Kwanjung Yee received a B.S. degree in Aerospace Engineering from Seoul National University in 1992. He then went on to receive his M.S. and Ph.D. degrees from Seoul National University in 1992 and 1998, respectively. Dr. Yee is currently an Assistant Professor at the Department of Aerospace Engineering in Pusan National University, Korea. His major research area covers unsteady aerodynamics, rotorcraft flight dynamics and multidisciplinary design optimization.



Sejong Oh received his B.S. and M.S. degrees in Aerospace Engineering from Seoul National University in 1979 and 1982 respectively. He finished his Ph.D. degrees from Stanford University in 1988. Prof. Oh is currently a Professor at Department of Aerospace at Pusan National University in Busan, Korea. He is currently a Board Member of Korean Society of Aeronautical and Space Sciences. Prof. Oh's research interests are in the area of Vortex flow and Rotorcraft aerodynamics.



Hee Jung Kang received a B.S. degree in Aerospace Engineering from KAIST in 1994. He then went on to receive his M.S. and Ph.D. degrees from KAIST in 1996 and 2001, respectively. Dr. Kang is currently a senior research engineer at Rotor Department in KARI (Korea Aerospace Research Institute). Dr. Kang's research interests are in the area of helicopter rotor aerodynamics, CFD (Computational Fluid Dynamics) and simulation of fluid-structure interaction.